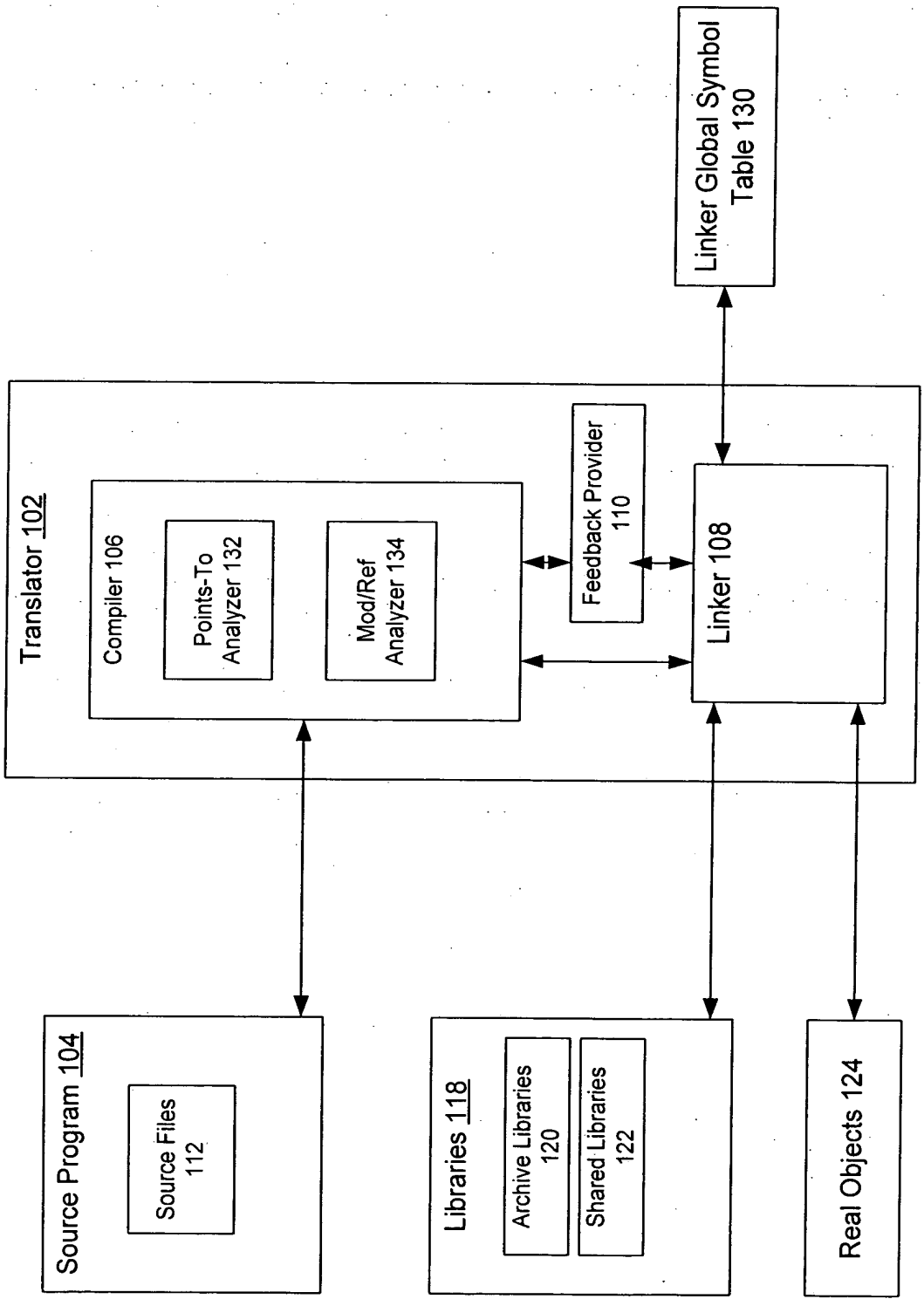
Linker Global Symbol Table 130
Translator 102
Compiler 106
Points-To Analyzer 132
Mod/Ref Analyzer 134
Feedback Provider 110
Linker 108
Source Program 104
Source Files 112
Libraries 118
Archive Libraries 120
Shared Libraries 122
Real Objects 124

FIG. 1

100

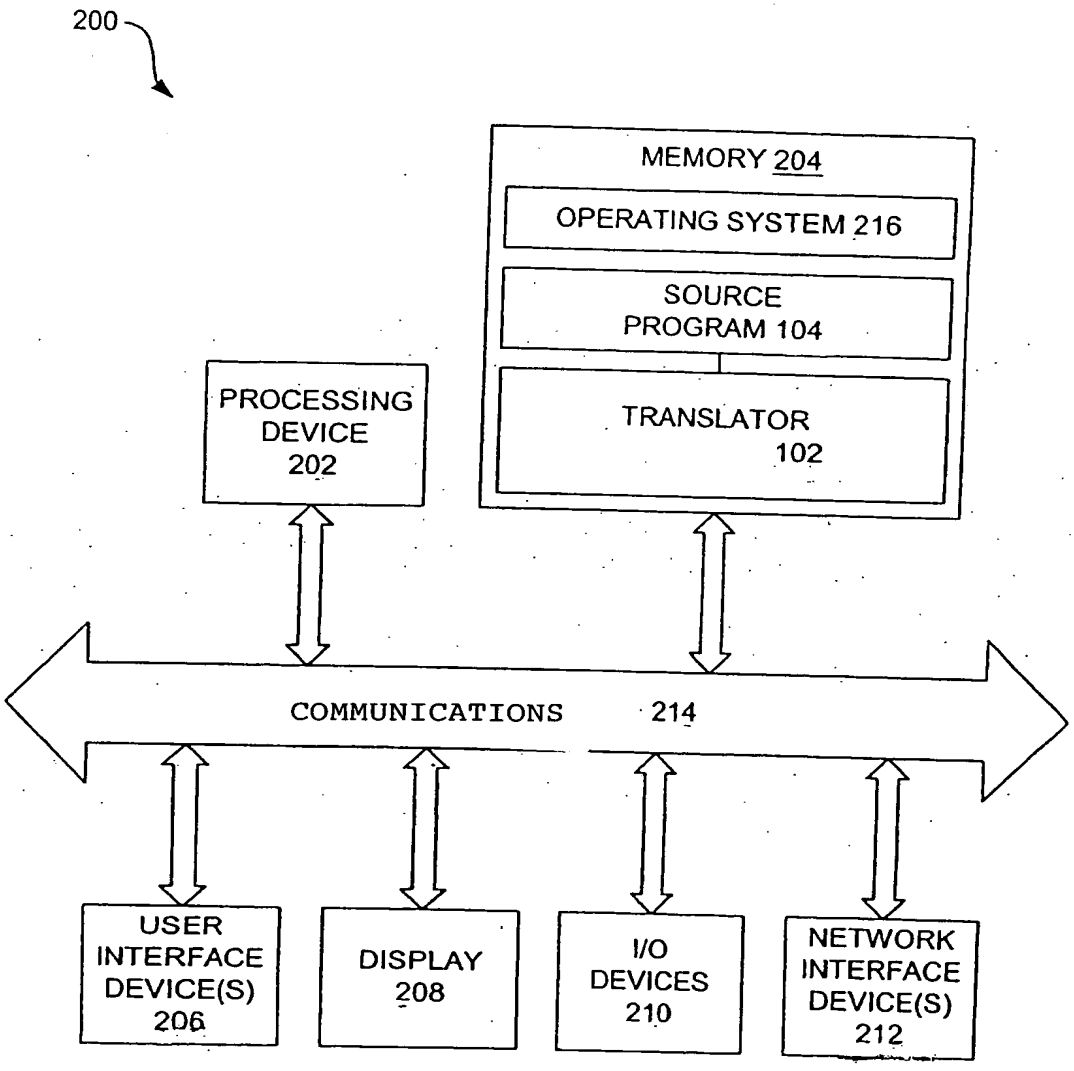
200
MEMORY 204
OPERATING SYSTEM 216
SOURCE PROGRAM 104
TRANSLATOR 102
PROCESSING DEVICE 202
COMMUNICATIONS 214
USER INTERFACE DEVICE(S) 206
DISPLAY 208
I/O DEVICES 210
NETWORK INTERFACE DEVICE(S) 212

FIG. 2

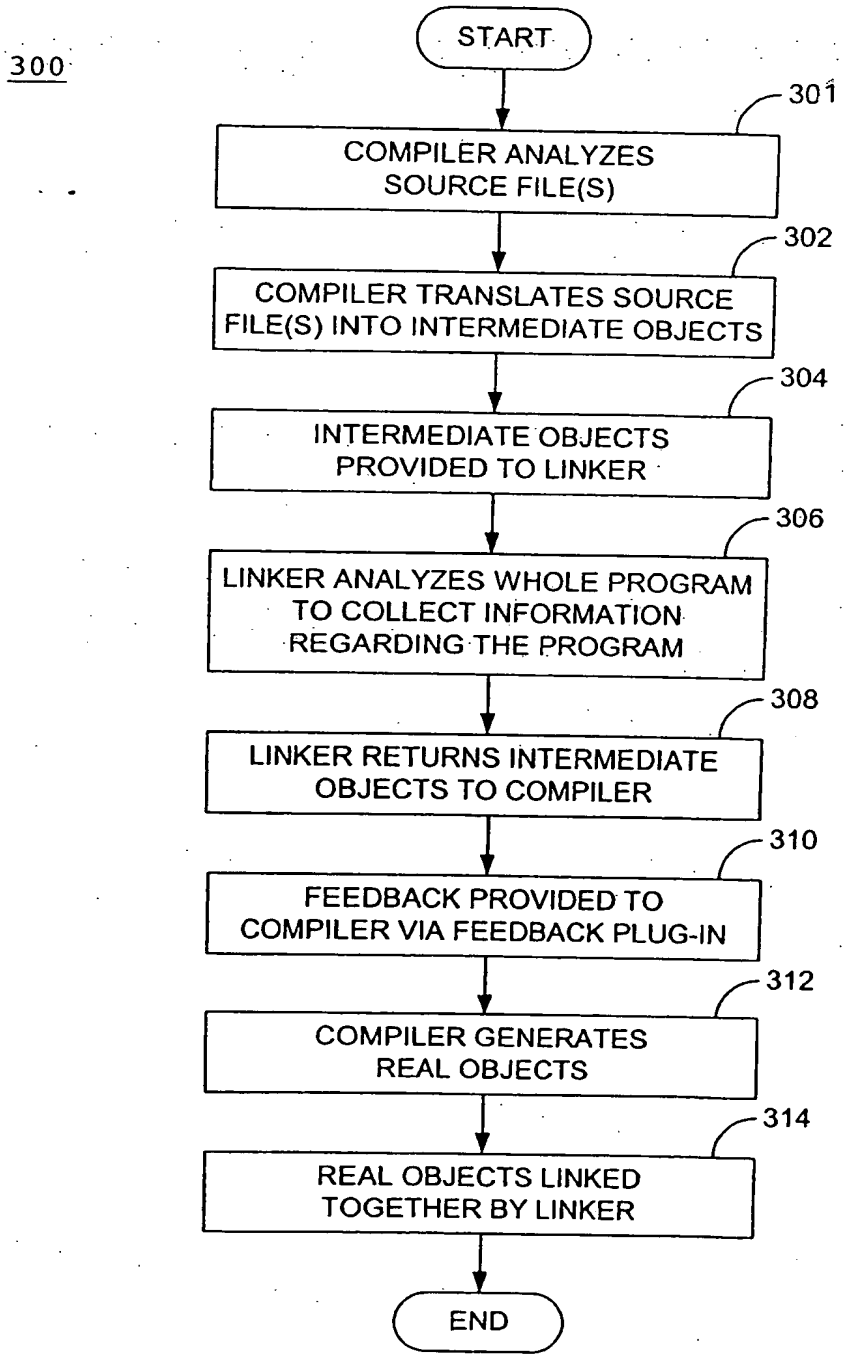
300
START
301
COMPILER ANALYZES SOURCE FILE(S)
302
COMPILER TRANSLATES SOURCE FILE(S) INTO INTERMEDIATE OBJECTS
304
INTERMEDIATE OBJECTS PROVIDED TO LINKER
306
LINKER ANALYZES WHOLE PROGRAM TO COLLECT INFORMATION REGARDING THE PROGRAM
308
LINKER RETURNS INTERMEDIATE OBJECTS TO COMPILER
310
FEEDBACK PROVIDED TO COMPILER VIA FEEDBACK PLUG-IN
312
COMPILER GENERATES REAL OBJECTS
314
REAL OBJECTS LINKED TOGETHER BY LINKER
END

FIG. 3

```
int g;

void bar(int** fp1, int** fp2, int i)
{

   *fp1 = &g;
   *fp2 = (int*) malloc(i*sizeof(int));
   return;
}


int foo()
{
   int* fp1, *fp2, i;

   bar(&fp1, &fp2, 100);
   g = 1;

   fp2[0] = 10;

   for  (i = 1; i< 100; i++)
   {
      fp2[i] = 2*g + 10;
   }

   *fp2+= 10;        (1)

   *fp1 = 20;        (2)

   return g;
}
```

FIG. 4

```
int foo()
{
   int *fp1, *fp2, i;
   bar(&fp1, &fp2, 100);

   fp2[0] = 10;

   for (i = 1; i< 100; i++)
   {
       fp2[i] = 12;
   }
   *fp2 += 10;
   *fp1 = 20;
   return 20;
}
```

FIG. 5

```
int * gp;

 int foo(int *f)
 {
      int *l = 0;

      l = f;    (1)    // <- this assignment need not be
represented in SIR

      ....

      gp = l;   (2)    // <-- this assignment to be
represented as gp = f in SIR.

 }
```

FIG. 6

```
struct A {
   int i;
   int* fd;
   ...
};

extern int g[]
extern int* gp;

int foo(A* f, int f2)
{
   A* lp;

   ....

   if (f2 > 10)
      lp = f->fd;
   else
      lp = (char*)&g + 12;


   gp = lp;   (1)

   ..
}
```

FIG. 7

```
int* gp;

int foo(int i)
{
    int* t = (int*) malloc(i); (1)

     ...
    gp = t;                    (2)

    bar();                     (3)
    ..
}
```

FIG. 8